Extreme Rainfall Prediction In India

Group 7

Debanjan Chatterjee - 20111016 (debanjan20@iitk.ac.in)
P J Leo Evenss - 20111038 (leoevenss20@iitk.ac.in)
Shilpa Chatterjee - 20111057 (shilpa20@iitk.ac.in)
Shruti Sharma - 20111061 (shruti20@iitk.ac.in)
Bopanna Tej Kiran - 20111070(tejkiranb20@iitk.ac.in)

Indian Institute of Technology,
Kanpur

2020

1 Abstract

Prediction of extreme rainfall is an important problem in the field of meteorology as it has an enormous impact on the life of people and a nation's economy. Every year people across the globe suffer from severe consequences of heavy rainfall like flood, spread of diseases,loss of life and belongings etc.

In this project we tried to predict extreme rainfall events 24 h to 48 h before the occurrence of the event using the previous climatic parameters. We tested various machine learning models on districts covering the entire Indian subcontinent. However, due to the problem of class imbalance, the model predicted several false positives and false negatives during classification. We tried to solve this issue through SMOTE.

2 Literature

Importance of Summer Monsoons to country's economy and its variability has motivated scientists to study these variations. They rigorously examine the weather patterns and deduce studies to prevent adverse effects of flood situations.
The 22 features considered in this study have been found to be important factors influencing rainfall.

3 Problem Statement

The available methods for heavy rainfall prediction are able to predict only 6 h prior to the event. We have tried to address the problem of predicting the occurrence of extreme rainfall events across regions during the summer monsoon months of June-September. The prediction is done based on the weather conditions over the region and it's surroundings in the past 48 h to 24 h. This will ensure that least damage is caused by heavy rainfall events.

For this purpose, we have used historical data of important weather variables that affect rainfall and trained our model. It is then tested with a new set of feature values to give a binary prediction about the extreme rainfall occurrence in the next 24–48 hrs.

4 Introduction

The science of climatic extremes is important and critical in terms of modeling, socio-economic impacts, damages, and adaptation.Though there has been significant research advancements in the field of meteorology, there are still notable number of extreme events resulting in huge human and

economic losses. One such event was Assam flood that started in May 2020 due to heavy rainfall affecting 30,000 and destroying crops across 5 districts.

As of October 2020, according to `https://en.wikipedia.org/wiki/2020_Assam_floods`, the floods affected over five million people, claiming the lives of 123 people, with an additional 26 deaths due to landslides, 5474 villages were affected and over one hundred and fifty thousand people found refuge in relief camps. This highlights the importance of early prediction of extreme rainfall.

5 Motivation

Heavy rainfall can occur due to some anomalous weather features prevalent in areas far away from the region of interest. We have therefore used the weather features over the entire Indian subcontinent. This will help in capturing the non-homogeneity in land-sea interaction, weather system and topography, over entire India, which affects rainfall in all parts of the country.

6 Dataset

We have obtained the weather parameters for the entire Indian subcontinent from National Centers for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR) reanalysis dataset. The latitude ranges from 5 degrees to 40 degrees north and longitude ranges from 65 degrees to 100 degrees east. In particular we have used daily data of weather parameters at surface and multiple levels from `https://psl.noaa.gov/data/gridded/data.ncep.reanalysis.surface.html`.

The NCEP/NCAR reanalysis data has a spatial grid resolution of 2.5° * 2.5°. The region is divided into 225 grids. So there are a total of 22*225 = 4950 variables available for each day since daily data is taken. For 48 h as well as 24 h prior prediction, the variables are increased to 4950*2 = 9900.

The surface level parameters are air temperature, mean sea level pressure, precipitable water, relative humidity, vertical wind velocity (omega), U-wind and V-wind. For multiple levels we have considered weather parameters at 850-, 600- and 400-hPa levels.The 850-,600-, 400-hPa level variables are air temperature, vertical wind velocity (omega), relative humidity, U-wind and V-wind. The surface and multiple level parameters sum up to 22 in total.

The weather parameters have been collected for the summer monsoon months in India i.e. June - September for the years 1997–2016.

The rainfall data is collected for the same time period from India Meteorological Department (IMD) (http://imdpune.gov.in/Clim_Pred_LRF_New/Grided_Data_Download.html)

7 Methodology

The validity of any statistical analysis depends on the quality of the data used in the analysis. Subdivisions are distributed fairly uniformly over the country, so districts from each subdivision were selected to form the network. A total of 9900 features were used.

There are 36 meteorological subdivisions of India. [1]

1. Andaman & Nicobar Islands
2. Arunachal Pradesh
3. Assam & Meghalaya
4. Naga., Mani., Mizo. & Tripura
5. Sub-Him. W. Bengal & Sikkim
6. Gangetic West Bengal
7. Orissa
8. Jharkhand
9. Bihar
10. East Uttar Pradesh
11. West Uttar Pradesh
12. Uttaranchal
13. Haryana, Chandigarh & Delhi
14. Punjab
15. Himachal Pradesh
16. Jammu & Kashmir
17. West Rajasthan
18. East Rajasthan
19. West Madhya Pradesh
20. East Madhya Pradesh
21. Gujarat
22. Saurashtra, Kutch & Diu
23. Konkan & Goa
24. Madhya Maharashtra
25. Marathwada
26. Vidarbha
27. Chattisgarh
28. Coastal Andhra Pradesh
29. Telangana
30. Rayalaseema
31. Tamil Nadu & Pondicherry
32. Coastal Karnataka
33. North Interior Karnataka
34. South Interior Karnataka
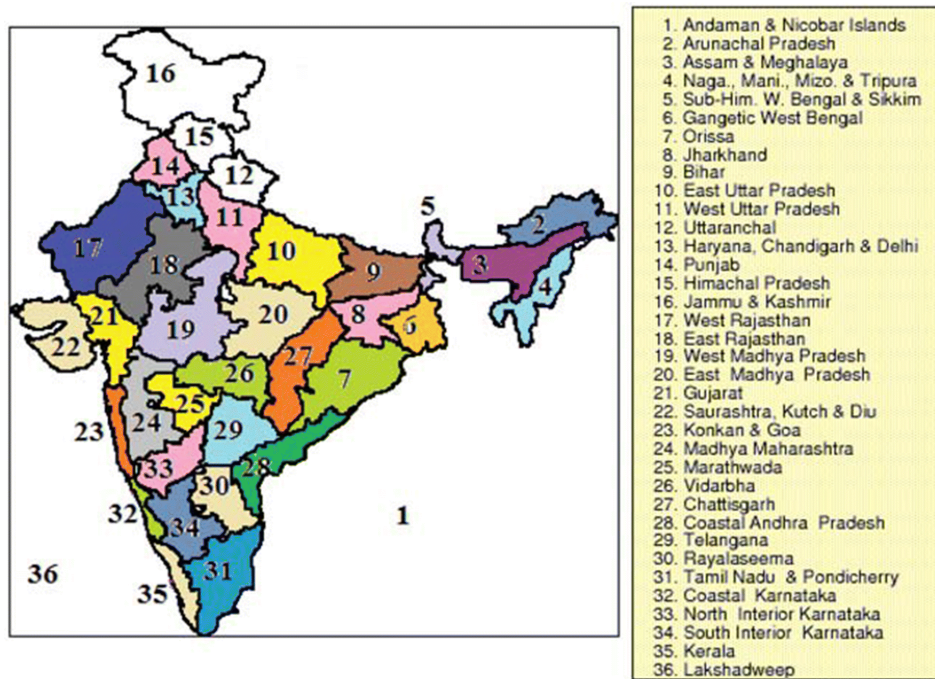35. Kerala
36. Lakshadweep

Figure 1

Missing values are flagged with -9.96921e+36f.

Threshold to determine extreme rainfall at a region = *Region's mean rainfall + Region's standard deviation from mean rainfall.*

We started with a set of 7 features namely air temperature, mean sea level pressure, precipitable water, relative humidity, vertical wind velocity (omega), U-wind and V-wind at the surface level. The Indian subcontinent

---

[1]Image Source: Meteorological Subdivision - OGD Platform India

3

was divided into equi-sized 225 grids and the daily data was prepared from these resulting in 225 * 7 =1575 features per day. The model trained with these features resulting in precision = 0.60 and recall = 0.16.

The features were then scaled up considering the different air pressure levels of 850-, 600- and 400-hPa and then dataset amounted to 22 feature attributes , thus 225 * 22=4950 per day and 4950 * 2=9900 for 2 consecutive days.

Feature Reduction

Method 1 : Auto-encoder- Such huge set of features if used for training a machine learning model may lead to overfitting. Therefore, we need a feature reduction technique. In this study, we want to learn the weather attributes which are mainly real-numbered and thus have chosen an auto-encoder architecture of deep learning for the purpose[?]. A simple auto-encoder is an unsupervised one layered neural network where the input X = $x_1, x_2, ..., x_n$ is a n dimensional feature vector. The output is given by (f is a non-linear transformation):

$$h_{W,b}(X) = f(W^T X) = f(\sum_{i=1}^{N} W_i x_i + b)$$

The hidden layer gets to learn a compressed representation of the input, such that the original input can be regenerated from it. The loss function is minimized through gradient descent.

Architecture -

- An input layer, (bias b = 9900)

- 2 hidden layers (initial units = 2500, bias = 2500),

- An output layer (bias = 9900)

- Activation function - Sigmoid

- Optimizer - Adagrad

- Learning rate - 0.1

- Batch size - 100

- Cost function - MSE

The first two layers make up an encoder while next two are the decoder. The shapes of input and output layers are same. The number of units in each layer has been set by trial and error. We multiply the output of input layer with a weight matrix and add biases, then take it through the sigmoid activation function and repeat the same for subsequent layers. The weights and biases get updated continuously as we train the model.

The size of encoded features comes out to be 2500 from initial 9900. This reduced, compact feature set can be used as input features for further processing.

Method 2 : Kernel PCA - RBF kernel was used. The feature set reduced to 2369 from 9900. However, it was experimentally found that the performance of autoencoder was superior to Kernel PCA.

Feature Scaling

Since the range of values of raw data varies widely, objective functions will not work properly without normalization. For example, many classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, range of all the features should be normalized so that each feature contributes approximately proportionately to the final distance.

Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it. And since, we used XGBoost which inherently uses gradient descent approaches we standardised our dataset using StandardScaler() from sklearn.

Class Imbalance

To effectively deal with the class imbalance problem in our biased dataset and to get the best performance, oversampling of the minority class has been tackled using SMOTE (Synthetic Minority Over-sampling TEchnique)[?] since features are continuous and we are solving a classification problem. In this new examples are synthesized from the existing examples.

SMOTE works by utilizing a k-nearest neighbor algorithm to create synthetic data. The procedure is repeated enough times until the minority class has the same proportion as the majority class.
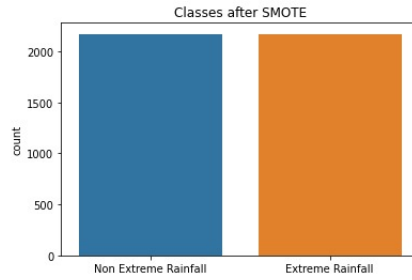
8 Results

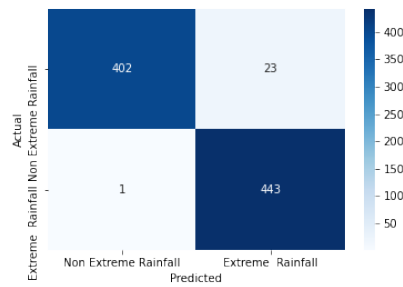### 8.1 Auto-encoder + SMOTE + XGB (24 h, Mumbai)

- High recall after dealing with all kinds of problems the data had originally, since Mumbai is one such city that receives heavy rainfall during summer monsoon months.
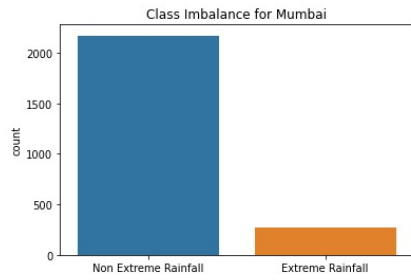
(a) Skewness in Mumbai Rainfall data

(b) After applying SMOTE

(a) Confusion Matrix

Accuracy– 0.9447640966628308
Precision– 0.9125
Recall– 0.9864864864864865
ROC_AUC score– 0.9438314785373608
F1–score– 0.948051948051948

(b) Metrics

8.2 Auto-encoder + XGB (48 h, Mumbai)

- Low recall because of class-imbalance. Also 24 h prediction performs better than 48 h.
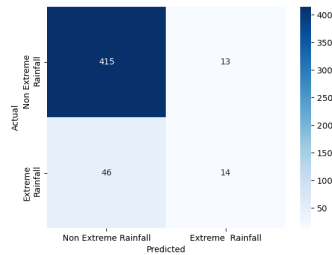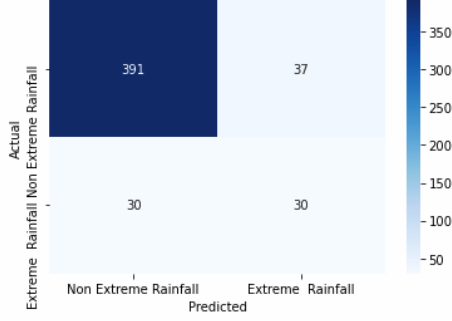

Figure 4: Confusion Matrix

Other Metrics

- Accuracy - 0.87
- Precision - 0.51
- Recall - 0.23
- F1-score - 0.32

8.3 XGB (24 h, Mumbai)

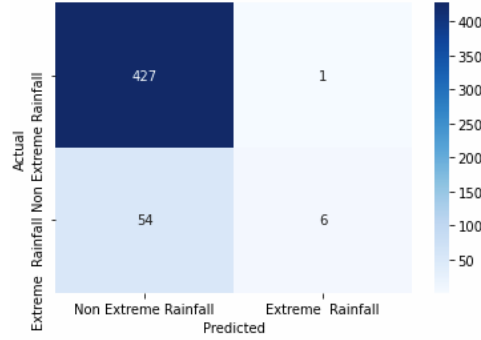- Low recall because no preprocessing done to balance the dataset.

(a) Confusion Matrix (XGB)

Accuracy- 0.8627049180327869
Precision- 0.44776119402985076
Recall- 0.5
ROC_AUC score- 0.7067757009345794
F1-score- 0.4724409448818898

(b) Metrics

8.4 SVM

- RBF kernel used.

(a) Confusion Matrix (SVM)

Accuracy- 0.8872950819672131
Precision- 0.8571428571428571
Recall- 0.1
ROC_AUC score- 0.5488317757009346
F1-score- 0.17910447761194032

(b) Metrics

9 Discussion

In this work we have explored techniques to learn and represent weather features and use them to predict extreme rainfall events. This model works better because we include all the features and try to understand underlying patterns and dependencies unlike other approaches which rely on selective feature reduction.

- For the country as a whole, the summer monsoon rainfall do not show any significant trend. However, there are large variations at the regional scale.

- Increasing parameters from 7 to 22 improved the quality of results to a great extent.

- 24 h prediction gave better results on average than 48 h.

- SMOTE gave excellent results.

- For model without SMOTE, even though a cost-sensitive model was used, the performance of model degraded, especially for cities with low percentage of extreme rainfall days, and thus made the dataset more skewed.

10   Future Direction

Here we have solved only a classification problem where we are only able to predict whether there will be heavy rainfall or not. In future we would also like to predict the amount of rainfall as well with the improved methods.

References

[1] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell.

[2] Goswami, B.N., Venugopal, V., Sengupta, D., Madhusoodanan, M., Xavier, P.K.: Increasing trend of extreme rain events over India in a warming environment.

[3] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks.

[4] Sulagna Gope, Sudeshna Sarkar, Pabitra Mitra, and Subimal Ghosh : Early Prediction of Extreme Rainfall Events: A Deep Learning Approach